

HT32 CMSIS-DSP Library 使用指南

文件编号: AN0538SC

简介

CMSIS 是由 ARM 制定的一个软件的标准接口, CMSIS 全名为“Cortex 微控制器标准软件接口(Cortex Microcontroller Software Interface Standard)”, 通过标准接口开发人员可以使用相同的接口控制不同供货商的微控制器, 进而缩短开发人员的开发与学习时间, 详细内容请参考 CMSIS 官方网站 <http://www.keil.com/pack/doc/CMSIS/General/html/index.html>。本文主要说明 CMSIS-DSP 应用在 HT32 系列微控制器, 包含环境建置、使用说明等内容。

功能说明

CMSIS-DSP 特点

CMSIS-DSP 是 CMSIS 的一个子项目, 其特点包括:

1. 专门给 Cortex-M 一套通用的信号处理函数。
2. ARM 提供的函数库拥有超过 60 个函数。
3. 支持包括 q7、q15、q31^(Note)以及 floating-point(32 bit)数据类型态。
4. 针对 Cortex-M4/M7/M33/M35P 对 SIMD 指令实现并优化。

Note: 函数库使用命名为 q7、q15、q31, 分别代表 8、16、32-bit 定点数。

CMSIS-DSP 函数库项目

CMSIS-DSP 函数库可分类为下列几项:

1. 基础数学函数\快速数学函数\复杂数学函数。
2. 信号过滤函数。
3. 矩阵函数。
4. 转置函数。
5. 马达控制函数。
6. 统计函数。
7. 支撑函数(Support functions)。
8. 插值函数(Interpolation functions)。

环境建置

本章节介绍范例中使用到的硬件与软件。

硬件

CMSIS-DSP 支持 HT32 全系列，但建议用户选用 SRAM 大于 4KB 的 MCU，原因是使用 CMSIS-DSP 的范例需要较大的 SRAM。本文以 ESK32-30501 作为举例，ESK32-30501 所使用的 MCU 为 HT32F52352。

软件

使用范例前请先确认已在合泰官方网站下载最新的 Holtek HT32 Firmware Library，下载位置如图 1，下载后请将其解压缩。

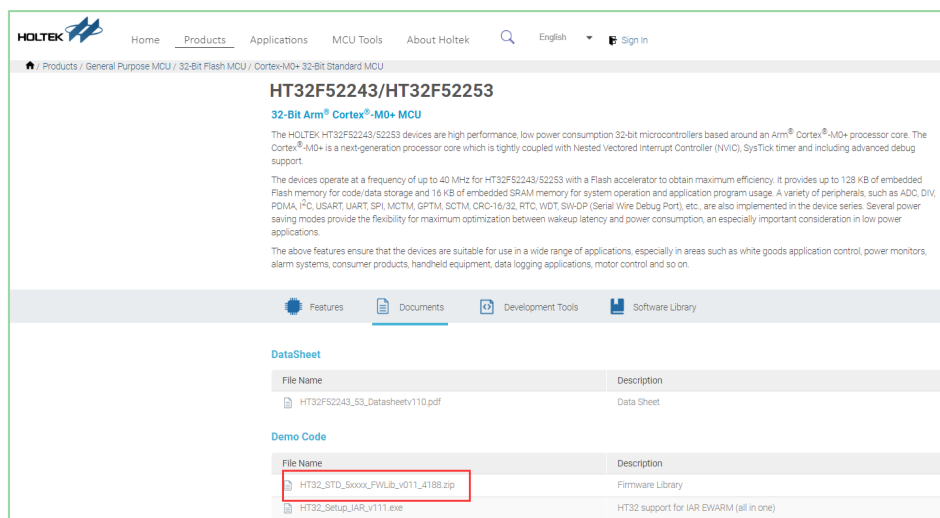


图 1、HT32 Firmware Library 下载网页

请通过下方链接下载 CMSIS-DSP 范例程序，此范例程序被打包成一个 ZIP 压缩文件，档名为 HT32_APPFW_xxxxx_CMSIS_DSP_vn_m.zip。

下载路径：http://mcu.holtek.com.tw/ht32/app.fw/CMSIS_DSP/

文件名格式说明请看下图 2：

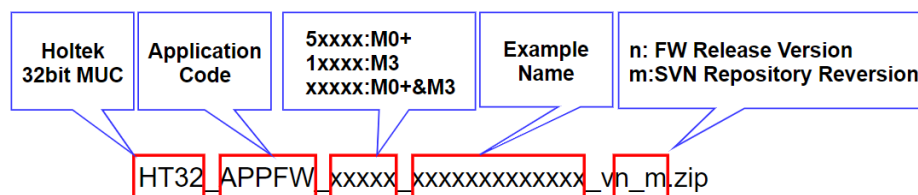


图 2、范例程序文件名介绍

Application Code 范例程序并不包含 Firmware Library 相关档案，因此在开始编译前，需将 Application Code 及 Firmware Library 解压缩的档案，放置到正确的路径。

Application Code 包含两个文件夹分别是 application 与 library，放置路径如下图 3，只需将这两个文件夹放到 Firmware Library 根目录下，就可以完成档案路径的配置，请参考图 4。您也可以选择将 Application Code 及 Firmware Library 两个压缩文件，同时解压缩到相同路径，此操作具有同样的效果。以此范例来说，解压缩后在 application 文件夹内会看到 CMSIS_DSP 的目录。

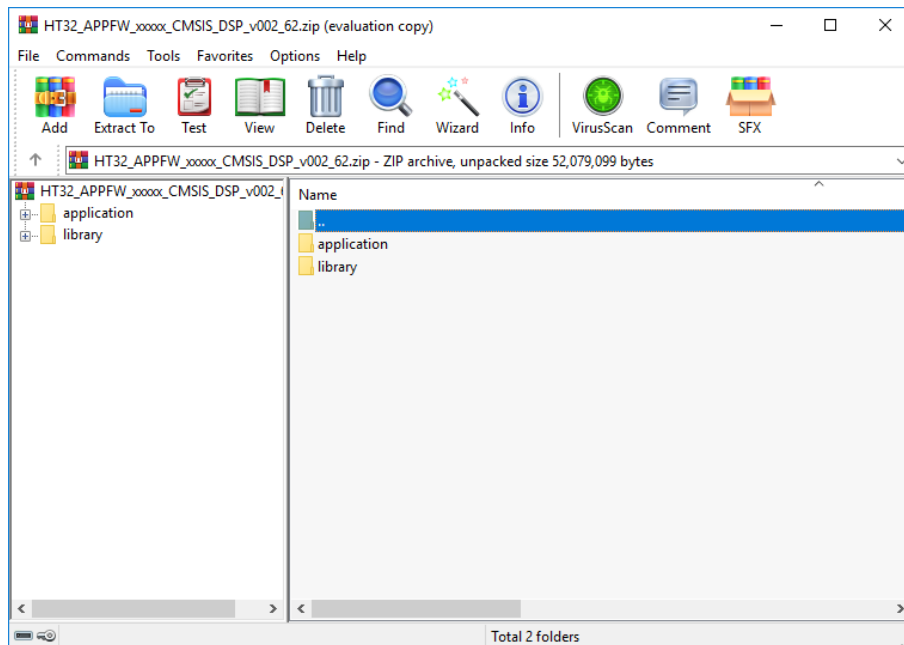


图 3、HT32_APPFW_xxxxx_CMSIS_DSP_vn_m.zip 内容

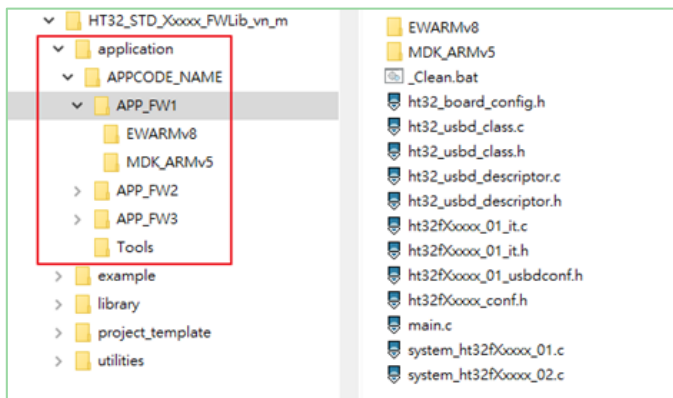


图 4、解压缩路径

档案结构

Application Code 内主要包含两个文件夹 library\CMSIS 与 application\CMSIS_DSP，各别说明如下。Application Code 内的 library\CMSIS 文件夹内容如下。

文件夹名称	说明
DSP_Lib	Application FW Source Code
DSP_Lib\Examples	内有多数 CMSIS-DSP 函数库的标准范例，由 ARM 提供，这些项目设定皆用模拟的方式执行，因此不需任何 MCU 就可执行程序，用户可以通过执行这些范例进而快速学习使用方法。
DSP_Lib\Source	CMSIS-DSP 函数库的源码。
Include	使用 CMSIS-DSP 函数库必需使用的头文件。

文件夹名称	说明
Include\arm_common_tables.h	外部(extern)数组变量的声明。
Include\arm_const_structs.h	外部常数声明。
Include\arm_math.h	此文件非常重要，此文件为使用 CMSIS-DSP 函数库的接口，调用任何函数库内 API 皆通过 arm_math.h。
Lib\ARM	给 ARMCC 的 CMSIS-DSP 函数库。 <ul style="list-style-type: none"> arm_cortexM3l_math.lib (Cortex-M3, Little endian) arm_cortexM0l_math.lib (Cortex-M0 / M0+, Little endian)
Lib\GCC	给 GCC 的 CMSIS-DSP 函数库。 <ul style="list-style-type: none"> libarm_cortexM3l_math.a (Cortex-M3, Little endian) libarm_cortexM0l_math.a (Cortex-M0 / M0+, Little endian)

application\CMSIS_DSP 内包含使用 HT32 系列 MCU 为平台的 CMSIS_DSP 相关范例，内有多个范例如下表，并支持 HT32 全系列 IC，开发项目为 Keil 的 MDK_ARM。

文件夹名称	说明
arm_class_marks_example	展示取得最大值、最小值、期望值、标准偏差、变异数和矩阵函数。
arm_convolution_example	通过复数型快速傅立叶变换(Complex FFT)和支撑函数展示回积分定理。
arm_dotproduct_example	展示通过向量的乘法与加法取得点积。
arm_fft_bin_example	展示通过复数型快速傅立叶变换、复数强度(Complex Magnitude)和最大函数模块来计算输入信号频域中最大能量窗口(bin)。
arm_fir_example	展示通过 FIR 实现低通滤波。
arm_graphic_equalizer_example	展示通过图形均衡器(Graphic Equalizer)来改变音质。
arm_linear_interp_example	展示线性插值模块和快速数学模块的用法。
arm_matrix_example	展示矩阵相关计算，包括转置矩阵、矩阵乘法和反矩阵。
arm_signal_converge_example	通过 NLMS(Normalized Least Mean Square)、FIR 和基础数学模型展示可自我调试的 FIR 低通滤波器。
arm_sin_cos_example	展示三角函数的计算。
arm_variance_example	展示通过基础数学和支撑函数计算变异数。
filter_iir_high_pass_example	展示通过 IIR 实现高通滤波。

测试

本文将以 application\CMSIS_DSP\arm_class_marks_example 作为测试范例，在测试前请确认 ESK32-30501 是否已接上 PC，并且 Application Code 与 Firmware Library 已在正确的位置。

请打开 application\CMSIS_DSP\arm_class_marks_example，执行 _CreateProject.bat，如下图，执行结束后请打开 MDK_ARMv5(若 Keilv4 请开启 MDK_ARM)，可以发现此范例支持全 HT32 系列，我们使用 ESK32-30501，因此开启项目 Project_52352.uvprojx。

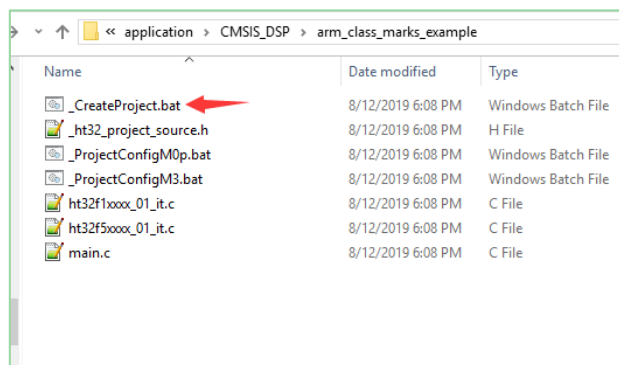


图 5、产生项目

开启项目后请编译(快捷键“F7”), 烧录(快捷键“F8”), 进 Debug(快捷键“Ctrl+F5”), 最后执行(快捷键“F5”)。执行的结果请观察下列变量。

变量名称	资料方向	说明	执行结果
testMarks_f32	Input	一个 20×4 的数组。	—
testUnity_f32	Input	一个 4×1 的数组。	—
testOutput	Output	为 testMarks_f32 与 testUnity_f32 的乘积。	{188, 229, 210...}
max_marks	Output	为 testOutput 数组内元素的最大值。	364
min_marks	Output	为 testOutput 数组内元素的最小值。	156
mean	Output	为 testOutput 数组内元素的期望值。	212.300003
std	Output	为 testOutput 数组内元素的标准差。	50.9128189
var	Output	为 testOutput 数组内元素的变异数。	2592.11523

使用说明

整合

本章节将介绍如何整合 CMSIS-DSP 到使用者的项目。

➤ 步骤 1:

首先在项目的设定内加入新的 Define 符号, M0 请加入“ARM_MATH_CM0PLUS”, M3 请加入“ARM_MATH_CM3”设定步骤为(1)Options of Target(快捷键“Alt+F7”), (2)选 C/C++的页签, (3)在 Define 加入新的定义。如下图。

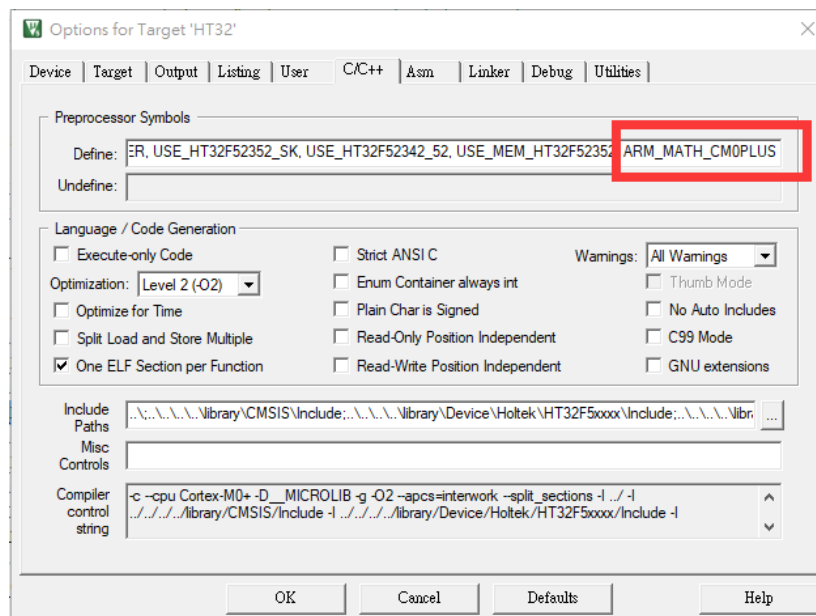


图 6、加新 define

➤ 步骤 2:

增加 Include 路径, 同样在 C/C++ 的页签, 请点选 Include Paths 旁边的按钮, 将弹出 Include 路径的窗口, 请增加新路径“..\..\..\Library\CMSIS\Include”。如下图。

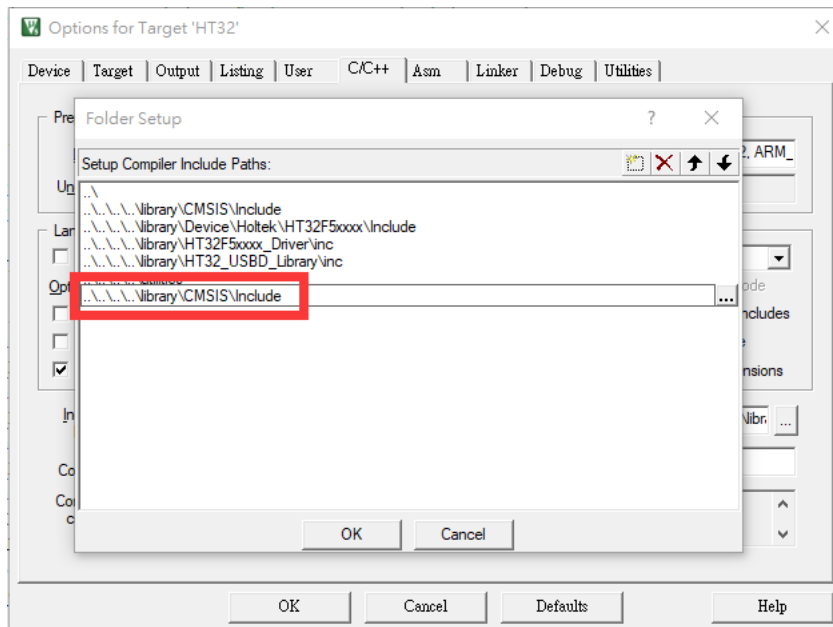


图 7、加新 include 路径

➤ 步骤 3(Optional):

加入函数库, 请点选按钮 Manage Project Items 如下图, 若是找不到此按钮可以点选 Window → Reset View to Defaults → Reset, 此时 IDE 会将窗口配置回到 Default 设定, 就可以找到按钮 Manage Project Item。

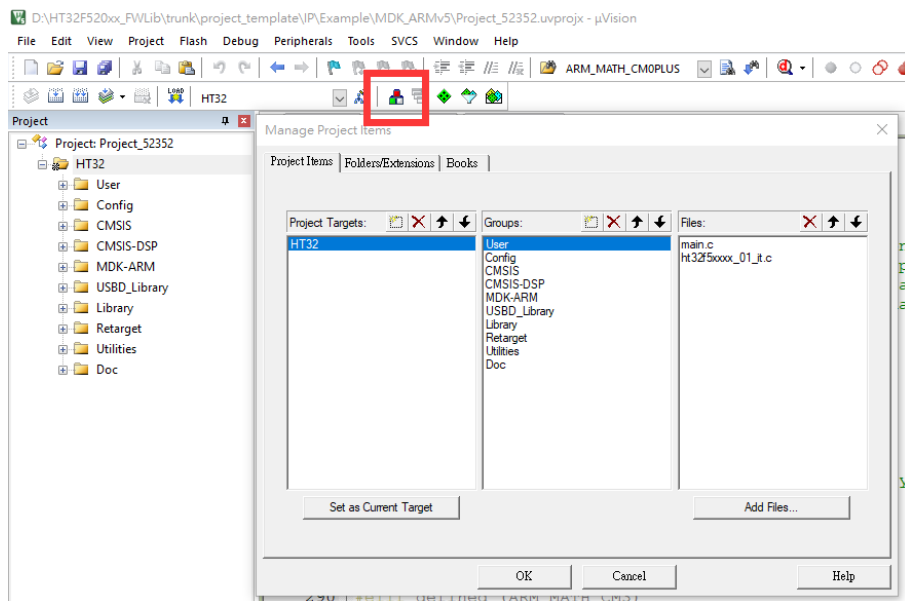


图 8、加新文件夹 1

通过下图红框内的功能增加“CMSIS-DSP”文件夹，并通过上移的功能将其移到 CMSIS 下方，完成后关闭 Manage Project Item。

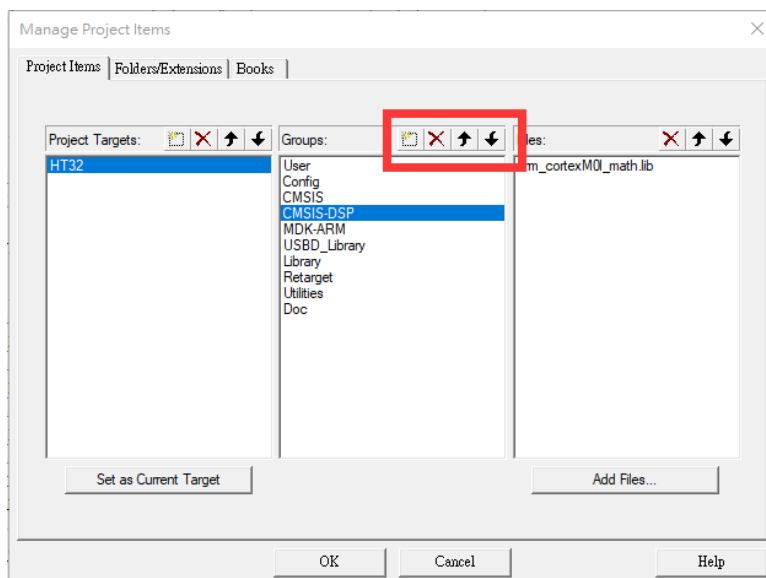


图 9、加新文件夹 2

➤ Step 4:

快速点击两下左边树形图的 CMSIS-DSP 文件夹(若是跳过 Step3, 请任意找个文件夹, 例如: User 或 CMSIS 等), 并加入 CMSIS-DSP 函数库, 如果是 M0+请选\library\CMSIS\Lib\ARM\arm_cortexM0l_math.lib, 若是 M3 请选\library\CMSIS\Lib\ARM\arm_cortexM3l_math.lib, 完成后会在 CMSIS-DSP 文件夹内发现函数库 arm_cortexMxl_math.lib, 如下图。

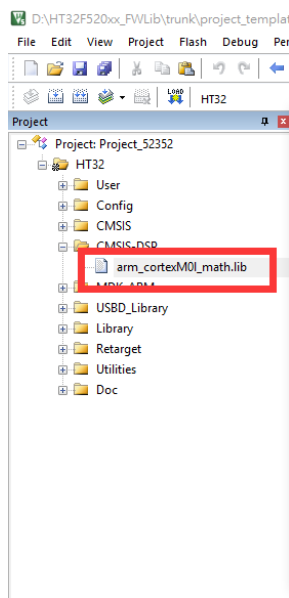


图 10、加新文件夹 3

➤ Step5:

最后请在 main.c 将头文件“arm_math.h”加入，如下图，此时完成所有的整合设定。

```

22  * the code, at their own risk. HOLTEK disclaims any expresse
23  * the warranties of merchantability, satisfactory quality an
24  *
25  * <h2><center>Copyright (C) Holtek Semiconductor Inc. All right
26  *****
27  // <<< Use Configuration Wizard in Context Menu >>>
28
29  /* Includes -----
30  #include "ht32.h"
31  #include "ht32_board.h"
32  #include "arm_math.h"
33
34  /** @addtogroup Project_Template Project Template
35  * @{}
36  */
37
38  /** @addtogroup IP_Examples IP
39  * @{}
40  */
41
42  /** @addtogroup Example
43  * @{}
44  */
45

```

图 11、加入“arm_math.h”

Low Pass Filter - FIR

此章节将介绍 application\CMSIS_DSP\arm_fir_example，通过此范例来展示如何设定 FIR 滤波器，并通过 FIR 滤波器移除高频信号。输入信号由 1kHz 的弦波与 15kHz 的弦波组成，信号的取样频率 48kHz，通过 FIR 低通滤波器将 6kHz 以上的信号移除，使其输出 1kHz 的信号。

范例的程序可以分成几个部分：

1. 初始化，初始化 FIR，使用到的 API 如下：

```
void arm_fir_init_f32 (arm_fir_instance_f32 *S, uint16_t numTaps, float32_t *pCoeffs,
float32_t *pState, uint32_t blockSize);
```

S: FIR 滤波器的结构。

numTaps: 滤波器的阶数(滤波器系数的个数)，此范例 numTaps =29。

pCoeffs: 滤波器的系数，此范例共 29 个，由 MATLAB 计算而得。

pState: 状态指标。

blockSize: 代表一次处理的样本数。

2. 低通滤波，通过调用 FIR 的 API，每次处理 32 个样本，总共处理 320 个样本，使用到的 API 如下：

```
void arm_fir_f32 (const arm_fir_instance_f32 *S, float32_t *pSrc, float32_t *pDst, uint32_t
blockSize);
```

S: FIR 滤波器的结构。

pSrc: 输入信号，此范例输入 1kHz 与 15kHz 的混合信号。

pDst: 输出信号，我们预期输出 1kHz 信号。

blockSize: 代表一次处理的样本数。

- 数据验证，通过 MATLAB 取得滤波的结果作为参考值，并通过 CMSIS-DSP 滤波的结果为实际值，两个结果来比较，目的为验证输出结果是否正确。

```
float arm_snr_f32(float *pRef, float *pTest, uint32_t buffSize)
```

pRef: 参考值，由 MATLAB 产生。

pTest: 实际值，由 CMSIS-DSP 产生。

blockSize: 代表一次处理的样本数。

执行结果如下图，Input 是过滤波器之前，Output 是经过滤波后的结果，Y 轴为信号的振幅，样本的取样频率是 48kHz，因此 X 轴数字加一，代表时间加 $20.833 \mu s(48kHz)$ 。从图 12 与图 13 比较可以发现 15kHz 被消除，只剩 1kHz 的信号。

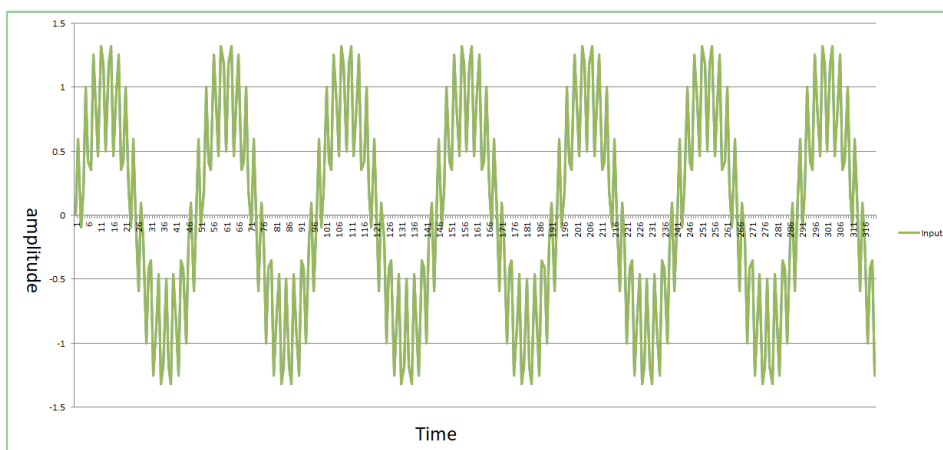


图 12、Input data

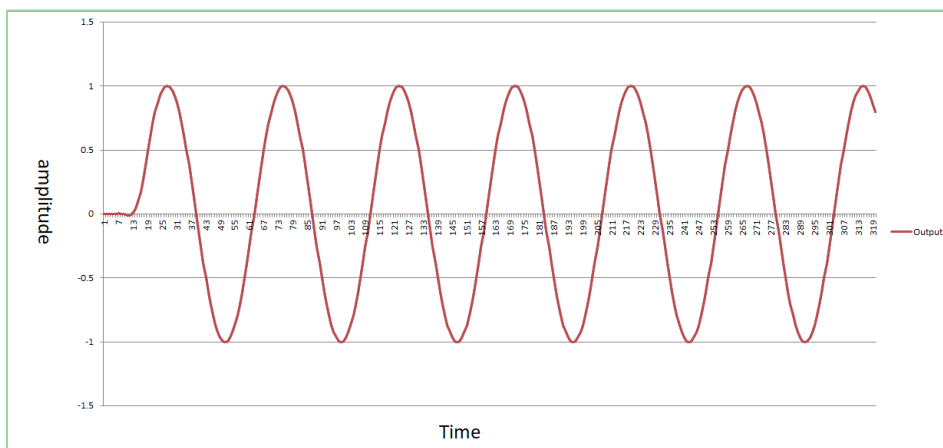


图 13、Output data

High Pass Filter - IIR

此章节将介绍 application\CMSIS_DSP\filter_iir_high_pass_example，通过此范例来展示如何设定 IIR 滤波器，并通过 IIR 滤波器移除低频信号。输入信号由 1Hz 的弦波与 30Hz 的弦波组成，信号的取样频率 100，总共取 480 个样本点，通过 IIR 高通滤波器将 7Hz 以下的信号移除。

范例的程序可以分成几个部分：

1. 产生 480 笔样本。样本 0 到 159 为 30Hz 的弦波，样本 160 到 319 为 1Hz 的弦波，样本 320 到 479 为 30Hz 的弦波，

2. 初始化，初始化 IIR，使用到的 API 如下：

```
void arm_biquad_cascade_df1_init_f32 (arm_biquad_casd_df1_inst_f32 *S, uint8_t numStages,
float32_t *pCoeffs, float32_t *pState));
```

S: IIR 滤波器的结构。

numStages: 过滤器中的二阶阶段数，此范例 numStages=1。

pCoeffs: 滤波器的系数，此范例共 5 个。

pState: 状态指标。

3. 高通滤波，通过调用 IIR 的 API，每次处理 1 个样本，总共处理 480 个样本，使用到的 API 如下：

```
void arm_biquad_cascade_df1_f32 (const arm_biquad_casd_df1_inst_f32 *S, float32_t *pSrc,
float32_t *pDst, uint32_t blockSize);
```

S: IIR 滤波器的结构。

pSrc: 输入信号，此范例输入 1Hz 与 30kHz 的混合信号。

pDst: 输出信号，我们预期输出只剩 30kHz 信号。

blockSize: 代表一次处理的样本数。

4. 结果输出，通过 printf 将输入信号与输出信号输出到 PC。

执行结果如下图，Input 是过滤波器之前，Output 是经过滤波后的结果，Y 轴为信号的振幅，样本的取样频率是 100Hz，因此 X 轴数字加一，代表时间加 10ms(100Hz)。从图 14 与图 15 比较可以发现 1Hz 被消除，只剩 30Hz 的信号。

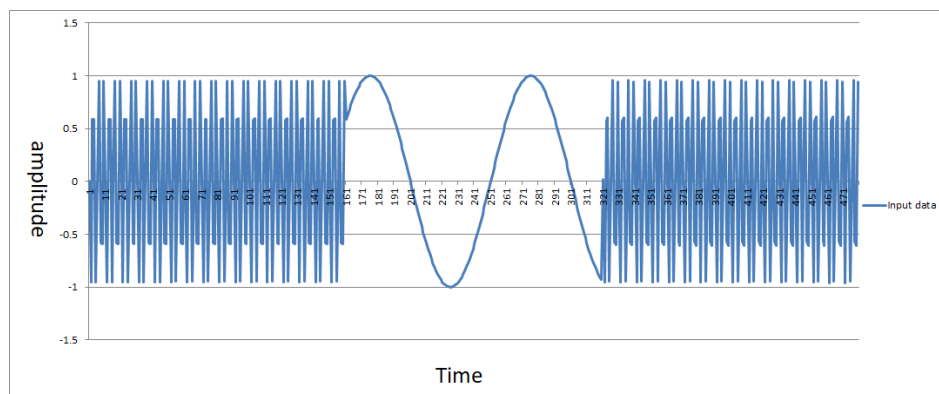


图 14、Output data

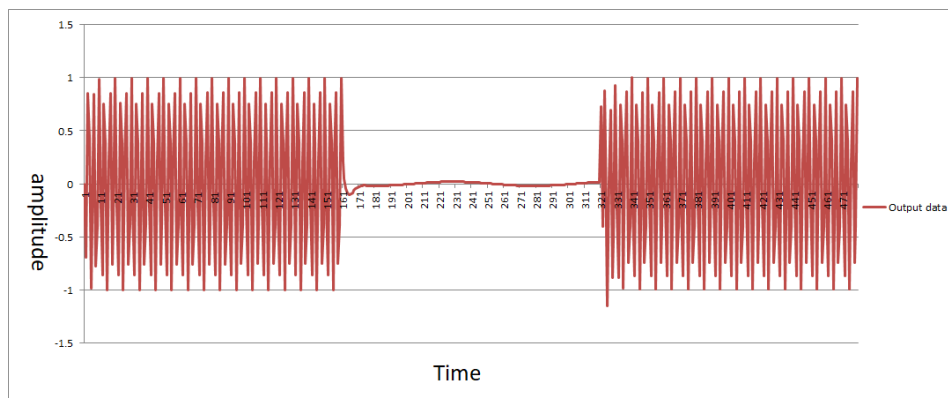


图 15、Output data

注意事项

使用 CMSIS-DSP 函数库时请特别注意编译后的内存大小，请在测试前确认内存没有溢出。

结论

CMSIS-DSP 对于信号的处理以及数学计算部分有强大的能力，建议诸位读者可善加利用。

参考资料

参考网站 <http://www.keil.com/pack/doc/CMSIS/General/html/index.html>。

版本及修改信息

Date 日期	Author 作者	Issue 发行
2019.09.03	王宏纶	第一版

免责声明

本网页所载的所有数据、商标、图片、链接及其他数据等 (以下简称「数据」), 只供参考之用, 盛群半导体股份有限公司 (以下简称「本公司」) 将会随时更改数据, 并由本公司决定而不作另行通知。虽然本公司已尽力确保本网页的数据准确性, 但本公司并不保证该等数据均为准确无误。本公司不会对任何错误或遗漏承担责任。

本公司不会对任何人士使用本网页而引致任何损害 (包括但不限于计算机病毒、系统故障、数据损失) 承担任何赔偿。本网页可能会连结至其他机构所提供的网页, 但这些网页并不是由本公司所控制。本公司不对这些网页所显示的内容作出任何保证或承担任何责任。

责任限制

在任何情况下, 本公司并不须就任何人由于直接或间接进入或使用本网站, 并就此内容上或任何产品、信息或服务, 而招致的任何损失或损害负任何责任。

管辖法律

本免责声明受中华民国法律约束, 并接受中华民国法院的管辖。

免责声明更新

本公司保留随时更新本免责声明的权利, 任何更改于本网站发布时, 立即生效。